straints placed on the system, model or parameters that must be met before the execution of the operation can begin. If precondition constraints are not met, a failure is generated by the constraint and the destructor never executes. A destructor descriptor describes a destructor with a set of attributes, but does not implement that constructor or know how to invoke that destructor. No additional events are added by the destructor descriptor.

[0167] A "metamodel" or a "model descriptor" of the present invention describes a model. A model descriptor of the present invention is a classifier descriptor. A model descriptor has a relationship with a version, a parent metamodel and a destructor descriptor. A model descriptor may have a relationship with the following elements as well: interface descriptors, static attribute implementations, instance attribute descriptors, static operation descriptors, instance operation descriptors, signal descriptors, and construction descriptors.

[0168] A model descriptor has a one-to-one aggregation relationship with a version that provides details about the number of modifications that have been made to the model. A model descriptor has a one-to-one association relationship with a parent metamodel that is the parent metamodel from which this metamodel of the present invention extends. If no created parent exists, the parent metamodel is Root Type. A model descriptor has a zero-to-many aggregation relationship with interface descriptors that describe the interfaces implemented by the model. Interfaces allow for polymorphic functionality in the program using the model. A model descriptor has a zero-to-many aggregation relationship with static attribute implementations that only occur once across all instances of a model. The attribute is implemented and instantiated in the model implementation. A model descriptor has a zero-to-many aggregation relationship with instance attribute descriptors that describe data attributes of the model. Attribute descriptors imply the ability to get and set the value of the attribute unless otherwise constrained. Therefore, operations setting and getting the value(s) of an attribute should not be added as operation descriptors. Instance attributes occur once per instance of a model. A model descriptor has a zero-to-many aggregation relationship with static operation descriptors that describe the functionality of the model. Static operations do not require an instance of the model in order to be executed and may only access static attributes of the model. A model descriptor has a zero-to-many aggregation relationship with instance operation descriptors that describe functionality of the model. Instance operations must be performed using an instance of the model and may access both instance and static attribute values. A model descriptor has a zero-to-many association relationship with signal descriptors that describe notifications the model produces and the mechanism for registering to receive these notifications. A model descriptor has a zero-to-many aggregation relationship with constructor descriptors that are the descriptors of constructors used to create instances of model implementations. A model descriptor has a one-to-one aggregation relationship with a destructor descriptor that is a descriptor of the destruction process to dispose of an instance of the implementation of this model.

[0169] A metamodel describes a model, but does not implement that model or know how to access the various features of a specific model implementation. No additional events are added by the metamodel.

[0170] An interface descriptor of the present invention is a classifier descriptor that describes an interface. An interface descriptor has a relationship with a version and may have a relationship with the following elements: other interface descriptors, static attribute implementations, and instance operation descriptors. An interface descriptor has a one-to-one association relationship with a version that provides details about the number of modifications that have been made to the model. An interface descriptor has a zero-to-many aggregation relationship with interface descriptors that describe the interfaces of the present invention extended by this interface. Interfaces allow for polymorphic functionality in the program using the interface. An interface descriptor has a zero-to-many aggregation relationship with static attribute implementations: All attributes defined in an interface have to be implementations and must be assigned an initial value. This value cannot be changed once set. An interface descriptor has a zero-to-many aggregation relationship with instance operation descriptors that describe functionality the interface understands how to perform. Instance operations must be performed using an instance of a model implementing the interface.

[0171] An interface descriptor describes an interface, but does not implement that interface or know how to access the various features of a specific interface implementation. No additional events are added by the interface descriptor.

[0172] A package descriptor of the present invention is a descriptor describing the logical grouping of models into packages. A package descriptor may contain zero or more packages and zero or more models that are described by the package descriptor. A package descriptor has a relationship with a name and may have a relationship with a display name, a description, attribute descriptors, access constraint descriptors, hints and roles.

[0173] A package descriptor has a one-to-one association relationship with a name attribute that provides a name for the package. Any tool using the package uses this name for the package. A package descriptor has a zero to one association relationship with a display name that provides a name to display to the user. The display name may be presented to human users as a more attractive alternative to the name. A package descriptor has a zero to one association relationship with a description that provides details about the parameter for its correct use. A description is mainly useful for human users and automated documentation. A package descriptor has a zero-to-many aggregation relationship with attribute descriptors. A package generally contains an attribute to hold child package implementations and an attribute to hold model implementations. Additional attributes may be added for attributes such as specification version, specification vendor, and so on. A package descriptor has a zero-to-many association relationship with access constraint descriptors that describe the security constraints placed upon access to a destructor. A package descriptor has a zero-to-many association relationship with hints that are name-value pairings, which add details to the metamodel that cannot adequately be captured anywhere else and that are inherited from feature descriptor. A package descriptor has a zero-to-many association relationship with roles that are each a group of related hints found to be commonly occurring.